INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 14, 609–625 (1992)

# PREDICTION OF THREE-DIMENSIONAL GENERAL SHAPE EXTRUDATES BY AN IMPLICIT ITERATIVE SCHEME

VINCENT LEGAT

*Applied Mechanics, Université Catholique de Louvain, Place du Levant, 2, B-1348 Louvain-la-Neuve, Belgium*

AND

JEAN-MARIE MARCHAL

*Polyflow S.A., Place de l'Université, 16, B-1348 Louvain-la-Neuve, Belgium*

SUMMARY

This paper presents a numerical technique for solving three-dimensional free surface problems in extrusion applications. The method is fully implicit in the sense that a Newton–Raphson scheme is applied on all variables, and geometrically general. In particular, the die section shape may be complex and contains multiple corners; very few restrictions apply on the mesh generation because the method does not require the nodes to be located on straight lines (spines). A clear distinction is introduced between the directions associated with the kinematic condition and the remeshing rules. As a difference with respect to earlier publications, these concepts are handled separately. Only Stokes problems are solved in this paper and we have not introduced surface tension. Therefore corners in the die section propagate discontinuities in the extrudate shape, an a method for relocating corners without losing the quadratic convergence of the scheme is presented. Data structures used for the implementation are briefly discussed.

We present results on the extrusion of various profiles, including a rectangular die (a benchmark problem) and various complex sections containing multiple corners.

KEY WORDS 3D extrusion Moving boundaries Kinematic condition Remeshing Finite elements Free surfaces

## 1. INTRODUCTION

Several numerical techniques have been proposed over the last few years to solve three-dimensional free surface problems. These methods are based either on boundary elements methods[1,2] or on finite element formulations.[3,4] The goal of the present paper is to present a numerical technique based on finite elements which extends to complex three-dimensional geometries implicit techniques used to calculate two-dimensional free surfaces.[5] The iterative technique is called implicit when it couples the velocity–pressure unknowns to the position variables. In implicit techniques a Newton–Raphson scheme can be used to solve the system in order to avoid an outer iteration on the free surface position above the flow solver itself.

However, most two-dimensional techniques used for simulating free surface flows cannot be extended as such to three dimensions. In particular, the presence of corners in the die section will introduce a new difficulty linked to the presence of discontinuities in the extrudate. This difficulty has been recognized by Karagiannis *et al.*[3,4] In Reference 4, a technique based on pathlines has been introduced to relocate corners. This method has been successfully used to solve non-isothermal free surface problems where large deformations are inroduced by the temperature effects. However, the pathline method as described in Reference 4 requires explicit relocalizations

0271–2091/92/050609–17$08.50

*Received 28 December 1990*
*Revised 18 June 1991*

of the free surface before Newton–Raphson iterations. Our technique avoids those explicit iterations.

We introduce discontinuities of the normal direction in the formulation of the free surface problem itself. A finite element formulation is then consistently derived in Section 3. A model with multiple normals and multiple kinematic conditions is introduced. The technique is validated on the examples of Section 5.

In order to preserve the geometrical generality of our method (we want to calculate complex dies), we introduce in Section 4 a remeshing technique based on the distance between interior and boundary nodes. Its major advantage over the 'spine' techniques is that the constraint of moving nodes in a single direction is dropped.

A full Newton–Raphson scheme is used to solve the system. The number of additional variables and the increase in frontal width with respect to a fixed geometry calculation are small. Therefore solving a moving boundary problem does not require much more CPU time than the fixed boundary problem provided the flow non-linearities are solved by means of a Newton–Raphson scheme. The method is advantageous in terms of CPU time (not in memory) with respect to most decoupled techniques. For example, the CPU time of a single iteration with the current method is 5%–10% higher than the CPU time for a fixed domain iteration. Our scheme converges in four or five iterations ($10^{-5}$), whereas an explicit technique typically requires 10–20 fixed domain iterations to converge.

The last section present three examples, each involving one or several corners. Selection of position interpolation versus velocity–pressure interpolation is briefly discussed. In all cases the convergence of the results with mesh refinement has been checked.

## 2. BASIC EQUATIONS

Let us consider the isothermal flow of a Newtonian fluid in a three-dimensional geometry. All results presented in this paper have been obtained with a Newtonian fluid, although extending the method to generalized Newtonian fluids or non-isothermal flows is straightforward.

Let $\Omega$ be the flow domain of boundary $\partial\Omega$. $\partial\Omega$ is partitioned into

- $\partial\Omega_D$ the boundary part on which Dirichlet boundary conditions apply;
- $\partial\Omega_N$ the boundary part on which Neumann boundary conditions apply; for the sake of simplicity all Neumann boundary conditions will be supposed homogeneous;
- $\partial\Omega_F$ the free surface itself.

Let $\mathbf{v}$ and $p$ denote respectively the velocity and pressure fields defined on $\Omega$ and let $h$ denote a scalar kinematic degree of freedom defined on $\partial\Omega_F$, which describes the motion of the free surface. In the absence of inertia and body forces the free surface problem is formulated as follows.

Find $(\mathbf{v}, p, h) \in V \times P \times H$ such that

$$\nabla \cdot (\eta \bar{\nabla} \mathbf{v}) - \nabla p = 0 \quad \text{on } \Omega, \tag{1}$$

$$\nabla \cdot \mathbf{v} = 0 \quad \text{on } \Omega, \tag{2}$$

$$\mathbf{v} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega_F, \tag{3}$$

together with

$$\mathbf{v} = \hat{\mathbf{v}} \quad \text{on } \partial\Omega_D, \tag{4}$$

$$(\eta \bar{\nabla} \mathbf{v} - p\mathbf{I}) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega_N \cup \partial\Omega_F, \tag{5}$$

where $\bar{\nabla}\mathbf{v}$ denotes the symmetric part of the gradient of $\mathbf{v}$, $\eta$ is the fluid viscosity and $\mathbf{n}$ is the unit normal vector pointing out of the domain $\Omega$. The symbols $V$ and $P$ denote appropriate velocity and pressure function spaces defined on $\Omega$; $H$ is a function space for the displacement of the free surface defined on $\partial\Omega_F$.

Let us consider the boundary conditions on $\partial\Omega_F$ in more detail. Assigning the force on a part of the boundary is a valid boundary condition for a Stokes problem in a fixed geometry;[6] equation (3) can also be regarded as a Dirichlet datum in the normal direction. For a Stokes problem on a *fixed* domain, imposing (3) *together* with (5) on $\partial\Omega_F$ would define an ill-posed problem. Defining the flow boundary conditions in terms of (3) and (5) simultaneously on $\partial\Omega_F$ requires the introduction of a kinematic degree of freedom $h$ which describes the motion of the free surface in a given direction. However, the direction of motion of free surface nodes can be selected *a priori* (it does not change during the iteration process) as long as the free surface does not become tangent to this direction of displacement. If this happens, all data in the normal direction (force *and* velocity) can no longer be satisfied. Therefore the direction of the geometrical degree of freedom $h$ is a *non-tangent* direction. It must be noted that the non-linearity introduced by the dependence of $\Omega$ upon $h$ is a serious difficulty for proving existence and uniqueness theorems for problem (1)–(5).

As stated before, the direction $\mathbf{d}$ for the displacement of the free surface is defined *a priori* and is *fixed* during the iterative process, $h$ being the amplitude of the displacement in this direction (see Figure 1). However, the direction $\mathbf{d}$ is allowed to vary from node to node. *Physical* considerations allow us to select directions $\mathbf{d}$ in such a way that there is no possibility for the free surface to become tangent to $\mathbf{d}$ during the iterative process.

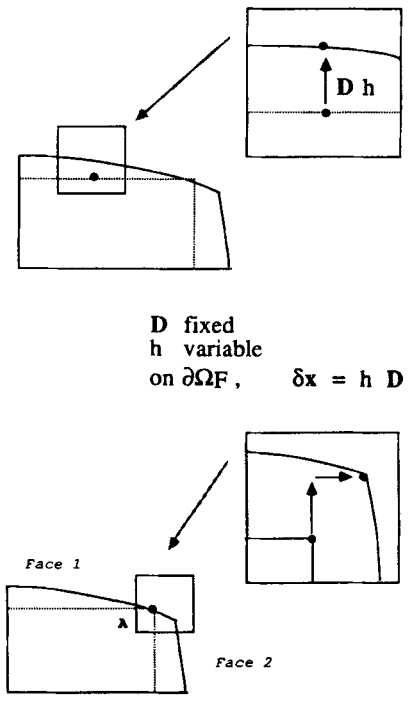


Figure 1. The geometrical degree of freedom $h$, the direction of displacement $\mathbf{d}$ and the corner strategy (multiple normals)

As in earlier publications on 2D or 3D free surface problems,[3–5] the selection of directions **d** for each node of the free surface is the first step in our solution technique; here **d** will be selected as the normal direction at a node in the initial (first-iteration) mesh (in a discrete sense). However, the **d**-direction does not influence the remeshing of the interior nodes, unlike the case of the spine tehniques where projections on the line of direction **d** are used to evaluate weights.

Most extrusion die sections contain sharp corners, as in Figure 1, and in the absence of surface tension the ability to properly handle corners is an essential ingredient of the simulation. The discontinuity of the slope in the cross-section propagates along the free surface itself and creates additional difficulties related to the definition of the normal, which have recently been recognized by Karagiannis *et al.*[4]

Our method relies on multiple normal definitions at singular points. Let us consider corner A in the die cross-section of Figure 1. Two normals are defined along the particle trajectory starting at point A in the lip section; one corresponds to face $F_1$ and the other to face $F_2$. If we consider a free surface $\partial\Omega_F$ consisting of $N_F$ 'faces' on which the normal is continuous, equation (3) will be written as

$$\mathbf{v}\cdot\mathbf{n}_i=0 \quad \text{on } \partial\Omega_{Fi}, \qquad 1\leqslant i\leqslant N_F. \tag{6}$$

Along lines of discontinuity of the normal, equation (6) implies that $\mathbf{v}\cdot\mathbf{n}$ vanishes on both faces. It must be pointed out that the presence of two kinematic conditions along some lines of the free surface is not generic to problem (1)–(5), in the sense that a single kinematic condition is the rule for all points on the free surface *except* for those points located along lines of discontinuity of the normal direction. The kinematic condition is a first-order transport equation for the free surface location (the position of the die lip being the initial condition) and discontinuities *can be transported* along characteristics. A problem in the formulation is expected only when the normal direction is discontinuous at *all points* on the free surface; this is obviously not the case in extrusion applications. Introducing discontinuities of the normal direction before finite element discretization shows that a discontinuity of **n** is compatible with the analytical formulation of the problem (no surface tension is included) and it will naturally come into the finite element formulation.

The double definition of normals along lines of discontinuity obviously has an influence on the selection of the geometrical degrees of freedom $h$ and the directions **d**. If one requires uniqueness of the normal direction on free surfaces, $h$ and **d** will now be partitioned as

$$h_i\in H_i \quad i\leqslant 1\leqslant N_F,$$

$$\mathbf{d}_i\in D_i, \quad i\leqslant 1\leqslant N_F,$$

where $H_i$ and $D_i$ are space of continuous functions defind on $\partial\Omega_{Fi}$.

The problem (1)–(5) will be reformulated as follows.

Find $(\mathbf{v}, p, h_i)\in V\times P\times\{H_i, 1\leqslant i\leqslant N_F\}$ such that

$$\nabla\cdot(\eta\bar{\nabla}\mathbf{v})-\nabla p=0 \quad \text{on } \Omega, \tag{7}$$

$$\nabla\cdot\mathbf{v}=0 \quad \text{on } \Omega, \tag{8}$$

$$\mathbf{v}\cdot\mathbf{n}_i=0 \quad \text{on } \partial\Omega_{Fi}, \qquad 1\leqslant i\leqslant N_F, \tag{9}$$

together with

$$\mathbf{v}=\hat{\mathbf{v}} \quad \text{on } \partial\Omega_D, \tag{10}$$

$$(\eta\bar{\nabla}\mathbf{v}-p\mathbf{I})\cdot\mathbf{n}=0 \quad \text{on } \partial\Omega_N\cup\partial\Omega_F. \tag{11}$$

## 3. FINITE ELEMENT DISCRETIZATION

The system (7)–(11) has been discretized by means of a standard finite element technique. Approximation spaces $V^h$, $P^h$ and $H_i^h$ $(1 \leqslant i \leqslant N_F)$ have been selected for the velocity, the pressure and the kinematic degree of freedom on each face.

On the basis of the Galerkin procedure the system (7)–(11) leads to the following discrete problem.

Find $(\mathbf{v}^h, p^h, h_i^h) \in V^h \times P^h \times \{H_i^h, 1 \leqslant i \leqslant N_F\}$ such that

$$\begin{aligned} \langle \eta \bar{\nabla} \mathbf{v} - p\mathbf{I}; \bar{\nabla} \mathbf{w} \rangle &= 0 \quad \forall \mathbf{w} \in V^h, \\ \langle \nabla \mathbf{v}; q \rangle &= 0 \quad \forall q \in P^h, \\ \langle \mathbf{v} \cdot \mathbf{n}_i; k_i \rangle &= 0 \quad \forall k_i \in H_i^h, \quad 1 \leqslant i \leqslant N_F. \end{aligned} \tag{12}$$

The symbol $\langle\ \rangle$ represents the $L_2$ scalar product on $\Omega$ or $\partial\Omega_F$; the space $V^h$ takes the Dirichlet boundary condition (4) into account and a zero displacement $h_i$ is prescribed at the lip of the extrusion die.

Triquadratic and trilinear functions on 'brick' elements have respectively been used for the velocity and pressure fields. For the geometrical degree of freedom $h_i$ we have used both biquadratic and bilinar shape functions. We have observed occasional divergences of the iterative scheme when a biquadratic representation is used in complex geometries for problems with sharp corners. However, for relatively simple geometries including a small number of corners (the rectangular section, for example), both quadratic and linear interpolations work. A comparison between bilinear and biquadratic free surface descriptions will be discussed in Section 5.

The system (12) is a non-linear system of equations in $(\mathbf{v}^h, p^h, h_i^h)$. All derivatives with respect to the velocity, pressure and position variables are calculated within a Newton–Raphson scheme. Since the system degenerates at the first iteration when the initial velocity field vanishes, the iterative scheme starts with a solution of the Stokes problem in a fixed geometry (generally a 'stick–slip' problem).

At this stage we have described a scheme for updating the boundary nodes; however, limiting the motion of the nodes to the sole free surface without updating the interior of the domain would lead to large element deformations in the vicinity of the free surface, which cannot be accepted in most simulations. There is a need for a *remeshing rule* which propagates the motion of the free surface in the domain interior. A major difference with respect to previous publications on free surface calculations for 2D problems[5] or 3D problems[3,4] is that *the remeshing rule has been kept totally independent* from the motion of the nodes on the free surface. In particular, we have not used a system of spines for relocating the interior nodes as in Reference 3.

It must be noted that some difficulties have been encountered in Reference 4 in a geometry involving sharp corners and that an outer iteration (above the Newton–Rahson iteration itself) was introduced for handling the remeshing at the corner; this leads to a loss of quadratic convergence of the iterative scheme.

## 4. REMESHING

In problems where the boundary nodes are relocated a rule must be selected for moving the interior nodes as a function of the boundary node positions with a view to an acceptable mesh deformation. Remeshing is naturally formulated as a boundary value problem and it is not surprising that most efficient techniques which map a parent geometry to a deformed one are

based on PDEs of elliptic type.[7] However, the problem has usually not been formulated as such in publications on three-dimensional free surfaces.

Saito and Scriven,[5] presented an implicit technique for computing two-dimensional free surfaces. The direction of extrusion is privileged; the domain is sliced in one-dimensional meshes on which a linear remeshing proportionality rule is applied. In a recent publication[3] a similar one-dimensional spine technique is used for calculating *three-dimensional* free surfaces; in three dimensions such a method requires us *to privilege two directions* for relocating the nodes on straight lines in place of one. This puts a severe constraint on the mesh generation, with additional difficulties at corners where the direction of displacement cannot be *a prior* imposed.[3,4]

Using PDEs as a remeshing rule is attractive; it would, however, considerably increase the cost of the calculation (and 3D Navier–Stokes simulations are CPU-intensive) because all nodal positions in the domain would be treated as independent variables. If we want to limit the number of geometrical variables to the elements of $H_i^h$ and still be able to derive a full Newton–Raphson scheme, we must select a rule where interior positions can be *a priori* computed as a *linear* function of the displacement on the boundary. All branching conditions must be excluded and corners strategies can generally not be used.

For extrusions problems we can privilege the direction of extrusion without loss of generality. Therefore we have introduced a 'slicing' of the domain after the exit of the die. 'Slicing' means that we introduce a system of planes normal to the direction of extrusion, every node belonging to one and only one plane (see Figure 2). The remeshing rule operates in these planes and is *two-dimensional.* Silicing the domain lowers the geometrical dimension of the remeshing rule and reduces the dependence of the Newton–Raphson matrices at a given node to the $h_i$ belonging to the section of this node only. A truly three-dimensional remeshing rule taking the whole boundary of the three-dimensional domain into account would increase the cost of the computation significantly because all geometrical variables would remain active throughout assembly of the system in a frontal solver.

For selecting a two-dimensional remeshing rule, three methods have been examined: a Thompson transformation;[7] an elliptic transformation based on a Laplace equation for which an elementary solution can be obtained (in two dimensions) by means of Green functions; an algebraic rule based on the Euclidean distance between the interior nodes and the boundary nodes.

### *4.1. Thompson transformation*

The Thompson transformation maps a 'parent' geometry described by co-ordinates $\mathbf{X}$ into a 'deformed' geometry described by co-ordinates $\mathbf{x}$ through the solution of the following problem:

$$\begin{aligned} \Delta_{\mathbf{x}}\mathbf{X} &= 0 \quad \text{on } \Omega_R, \\ \mathbf{x} &= \hat{\mathbf{x}} \quad \text{on } \partial\Omega_R. \end{aligned} \tag{13}$$

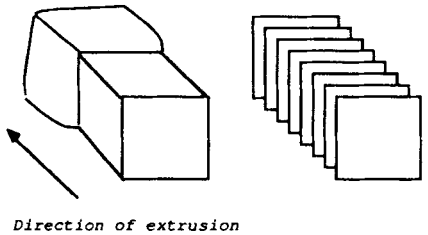


Figure 2. 'Slicing' of the computational domain in the direction of extrusion

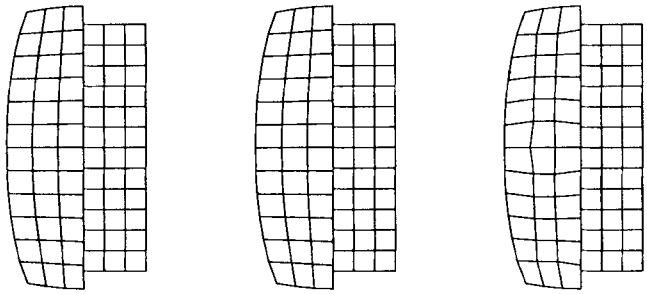


Figure 3. Remeshing of a rectangular die by (a) a Thompson transformation, (b) a Laplace transformation and (c) a Euclidean distance rule

$\Omega_R$ stands for the domain of each cross-section. $\Delta_x$ stands for the Laplace operator with respect to the co-ordinates $\mathbf{x}$; the position $\hat{\mathbf{x}}$ is prescribed on the domain boundary $\partial\Omega_R$. The fact that $\Delta_x$ operates on the deformed geometry makes the problem non-linear. The system (13) can be transformed into a set of non-linear PDEs in the co-ordinate system of the parent $\mathbf{X}$-geometry.[7]

The Thompson transformation has been tested on several two-dimensional sections and usually produces smooth grids (see Figure 3(a)). A major disadvantage is the non-linearity of system (13) written in the parent domain, for which an elementary solution cannot be found. All nodal positions must be treated as independent variables, while the number of variables is significantly increased with respect to a method involving ($\mathbf{v}^h$, $p^h$ and $h^h$) only. The method has been successfully tested in complex geometries, but we present only the results for the example described in Section 5.1.

### *4.2. Laplacian transformation*

In regular domains and for small mesh deformations the operator $\Delta_x$ can be replaced by $\Delta_X$, i.e. the Laplacian with respect to the co-ordinates in the *parent* domain. The system (13) then becomes

$$\begin{aligned} \Delta_X \mathbf{x} &= 0 \quad \text{on } \Omega_R, \\ \mathbf{x} &= \hat{\mathbf{x}} \quad \text{on } \partial\Omega_R. \end{aligned} \tag{14}$$

The elementary solution of (14) is a set of functions $\mathbf{w}$ defined on $\Omega_R \times \partial\Omega_R$ such that

$$\mathbf{x} = \int_{\partial\Omega_R} \mathbf{w}(\mathbf{x}, \hat{\mathbf{x}})\, d\hat{\mathbf{x}} \quad \forall \mathbf{x} \in \Omega_R.$$

The Green functions $\mathbf{w}$ evaluate the influence in the domain $\Omega_R$ of a Dirichlet datum on the domain boundary $\partial\Omega_R$.

Evaluating Green functions *a priori* requires a large number of computations; for two-dimensional domain $\partial\Omega_R$ (cross-sections) of sufficient regularity the Green functions can be easily evaluated by means of a conformal transformation.[8] It is not the objective of this paper to describe such a method in detail; let us say, however, that in its finite element discretization only two numerical solutions of a scalar Laplace problem on $\Omega_R$ are required to calculate the influence

of every boundary node on any interior node. These computations are typically performed before starting the iterative process.

Once the Green functions have been evaluated, all interior positions in a section can be eliminated as a function of the geometrical degrees of freedom $h_i^h$ of the section; see Figure 3(b) as an example of a deformed mesh based on an Laplace transformation. An important restriction is that the original mesh boundary must be $C^1$-continuous for evaluating correctly the Green function by means of a conformal transformation. With grids containing sharp corners this usually leads to numerical error and to artificially distorted meshes. The requirement on the continuity of the boundary is a major drawback of this remeshing technique, although it can be used successfully in domains with a smooth boundary.

### *4.3. Euclidean distance remeshing*

The concept at the origin of the method is that the closer an interior node becomes to a boundary node, the closer its displacement is related to the displacement of this boundary node. More precisely, the position of an interior node $\mathbf{x}_i$ is written as

$$\mathbf{x}_i^{\text{new}} = \mathbf{x}_i^0 + \sum_{j=1,N} w_{ij}(\mathbf{x}_j^{\text{new}} - \mathbf{x}_j^0), \tag{15}$$

where $j$ is an index describing the $N$ nodes of $\partial\Omega_R$ and $w_{ij}$ is a function of the Euclidean distance between node $i$ and node $j$:

$$w_{ij} = \frac{\sum_{k=1,N} |\mathbf{x}_i^0 - \mathbf{x}_k^0|_2}{N|\mathbf{x}_i^0 - \mathbf{x}_j^0|_2}.$$

An example of sections remeshed by the Euclidean distance rule is given in Figure 3(c). It has been observed that the Euclidean distance rule generally produces very good results in extrusion problems; the method has been selected for all applications described in Section 5. It must be pointed out that $w_{ij}$ could have been evaluated differently. For example, using Green functions (elementary solutions of a Laplace equation) for evaluating $w_{ij}$ also produces very good results. In this case the Laplace equation is used to evaluate variations of position, while rule (14) prescribes the position itself.

In order to smoothly deform sections without relocating nodes out of the planes of symmetry of the section, we have introduced a hierarchy in the geometrical dimension of the rules that we are using. This is the object of Section 4.4.

### *4.4. The remeshing cascade*

Using (15) requires knowledge of the motions of all nodes on $\partial\Omega_R$. This is usually not the case when some parts of the extrudate section boundary lie on planes of symmetry. In addition, there is also a need for a remeshing rule on the boundary $\partial\Omega_R$ itself, because the motion of the free surface $\partial\Omega_{R(\text{Face 1})}$ introduces deformations in the tangential direction on $\partial\Omega_{R(\text{Face 2})}$ (see Figure 4). Therefore one-dimensional remeshing rules have been introduced on boundary segments of $\partial\Omega_R$: such that points are relocated tangentially as a linear function of the motion of the extremities of the segment. This naturally introduces a hierarchy between 1D, 2D and 3D remeshing rules, where 1D positions on the boundary segments dictated by displacements of segment extremities act as boundary conditions for the 2D Euclidean distance rule, which are in turn data for relocating mid-side nodes located *between* the sections for trilinear co-ordinate fields (see Figure 4).

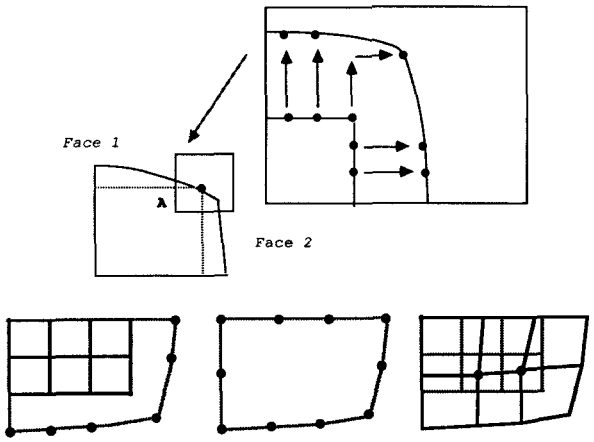


Figure 4. Remeshing rules along boundaries and the remeshing cascade

Hierarchy does not mean that the rules are applied sequentially but that constraints between co-ordinates and geometrical variables coming from a 1D (2D) rule are *preferred* to those coming from a 2D (3D) rule. Consequently, constraints of high priority are substituted in subsequent rules.

From a practical point of view, constraints between variables are manipulated symbolically and substitution occurs automatically once the hierarchy between constraints has been defined. This means that we execute linear combination on the lines and columns for each constrained variable during solution of the linear system. The code has been organized as follows.

1. At the level of the local matrices all nodal positions $\mathbf{x}^h$ are *variables*; this means that we do not substitute the geometrical variable $h_i^h$ as a primal unknown in the element matrices.
2. Linear relationships between internal and boundary node displacements are expressed as constraints; once a variable has been constrained, it *disappears* from the list of variables of the problem and a duality technique reports all derivatives with respect to the constrained variable on the variables which are constraining it.
3. On the free surfaces the displacement of the nodes is a function of the geometrical variable $h_i^h$ and the (known) direction of displacement $\mathbf{d}_i$.
4. Finally, the kinematic condition is a scalar PDE for the $h_i^h$.

This organization has a great flexibility in the sense that changing from one remeshing rule to another never requires one to modify the local finite element matrices. Writing the finite element matrices with position derivatives is the most tedious task in deriving a full Newton–Raphson scheme for problems with moving boundaries. Moreover, symbolic manipulation on the constraints is essential to the modularity of the program.

## 5. EXAMPLES

The code has first been validated on three-dimensional flat and circular dies for which a two-dimensional solution is easily calculated. We do not report these results here because they are essentially identical to well-known two-dimensional results. Furthermore, in order to verify the convergence of our method and to compare our results, we have considered the square die problem. We report in Table I the swelling ratio $Sw_m$ in planes of symmetry.

Table I. $Sw_m$ for the square die

| Authors | Unknowns | $Sw_m$ |
|---|---|---|
| Karagiannis *et al.* | 16500 | 18·4 |
| Wambersie and Crochet[9] | 6917 | 18·5 |
| Present work | 13616 | 18·5 |

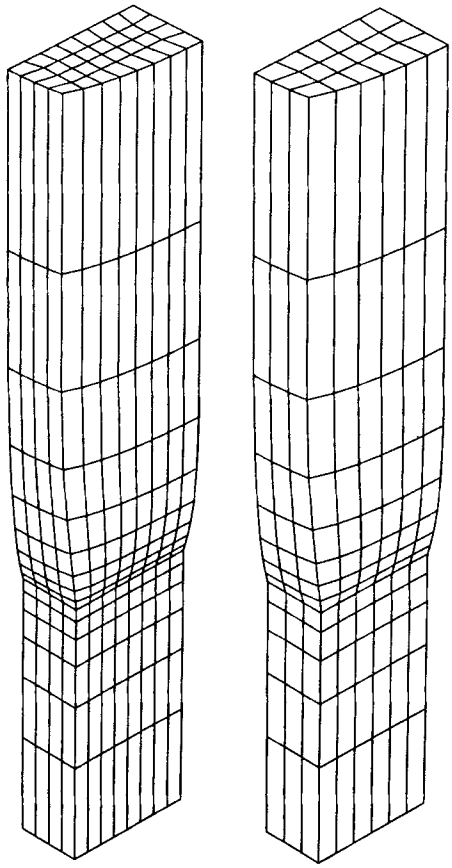

Figure 5. Perspective plot of the deformed meshes for the rectangular die

### *5.1. Rectangular die*

Extrusion of a Newtonian fluid through a rectangular die is a test problem for three-dimensional extrusion. The geometry is shown in Figure 5; symmetry of the problem with respect to he $x=0$ and $y=0$ planes has been used.

The rectangular die has been used as a test problem to validate the introduction of multiple normals at corners and to evaluate bilinear and biquadratic shape functions for the free surface position. In Figure 6 we have used a single normal at corner A, whose direction is computed by means of a least squares formulation. The direction of displacement of nodes located at the

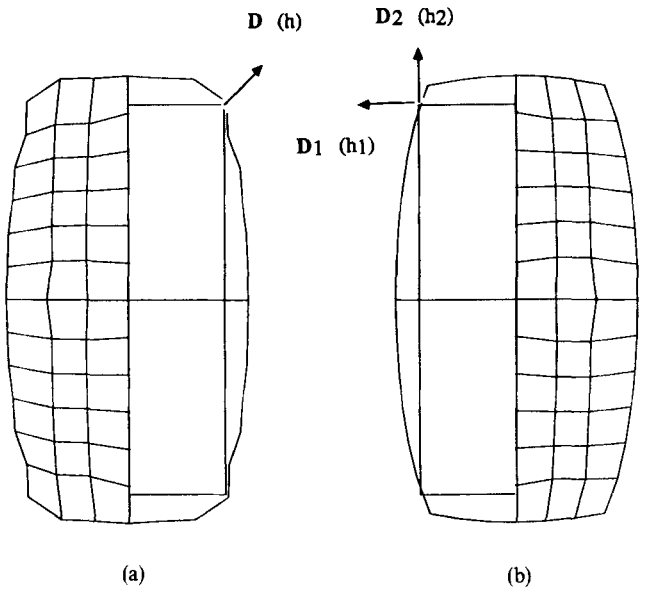


Figure 6. Profiles obtained by (a) single-normal model and (b) multiple-normal model

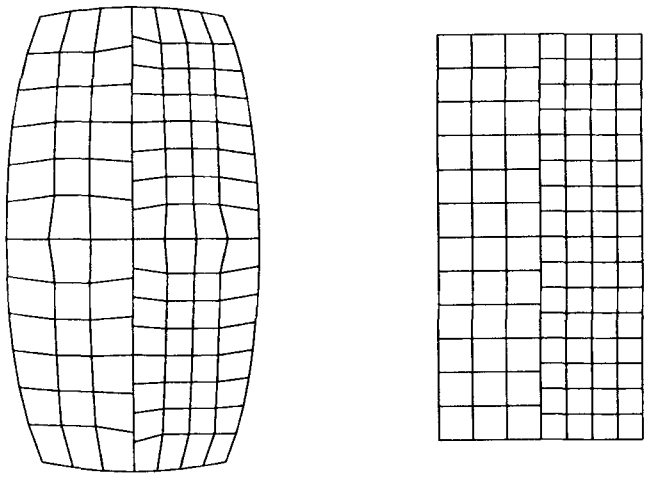

Figure 7. Rectangular die: mesh refinement analysis

corners is then prescribed. Since this direction does not in general correspond to the displacement of physical corners, the method will have a tendency to create a shape discontinuity at another place. Wiggles appear in the solution, as shown in Figure 6, or the Newton–Raphson scheme does not even converge. Solutions converged with mesh refinement could not be obtained with a single-normal model.

When singular points are allowed to move in two directions, the problem disappears and a solution confirmed with mesh refinement is shown in Figures 7 and 5. These solutions were

obtained with bilinear geometrical varibles; they converge at a relative precision of $10^{-6}$ in five or six iterations. Biquadratic shape functions have also been used for the geometrical variables.

### *5.2. Cross-like die*

Extrusion of a cross-like profile involving 12 corners is considered, as described in Figure 8. We have used the symmetry of the problem with respect to the $x=0$ and $y=0$ planes and have introduced a discontinuity of the normal direction along corners A, B and C in Figure 8.

Bilinear shape functions have been used for $H_i^h$ and quadratic convergence of the iteration scheme has been observed. We have also simulated the same cross die problem with biquadratic shape functions for $H_i^h$ with die discontinuity of the normals at corners. Although convergence of the algorithm would be obtained on some meshes, wiggles were observed in the extrudate shapes. These wiggles were not observed with bilinear shape functions. On some refined meshes divergence of the iterative scheme has been observed. In such cases the bilinear $h_i^h$, however, converged towards a smooth solution, which was coherent with results obtained with a coarse mesh. For this reason biquadratic $H_i^h$-fields have been abandoned in all subsequent simulations.

Two mesh refinements have been considered. Figure 8 presents cross-sections of these meshes at the die exit. A remeshing rule based on the Euclidean distance has been used in both cases. Quadratic convergence of the iteration scheme has been observed. Figure 9 presents perspective plots of the mesh boundaries and Figure 8 shows a comparison between the coarse and the fine

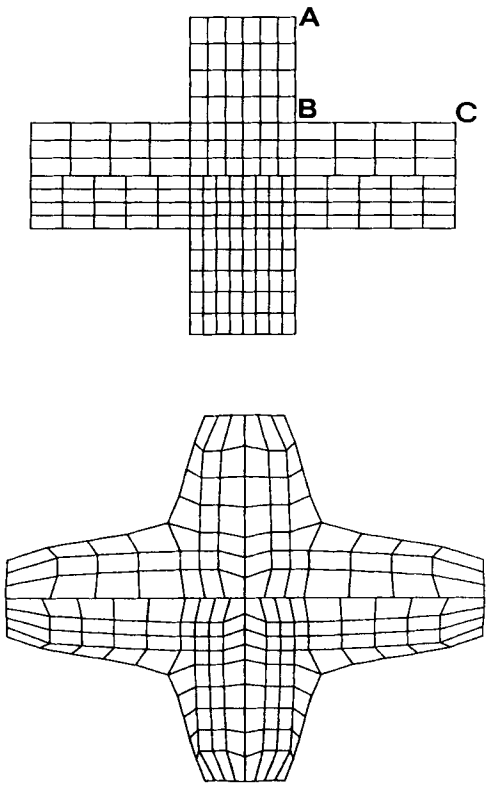


Figure 8. Cross-like die: mesh refinement analysis

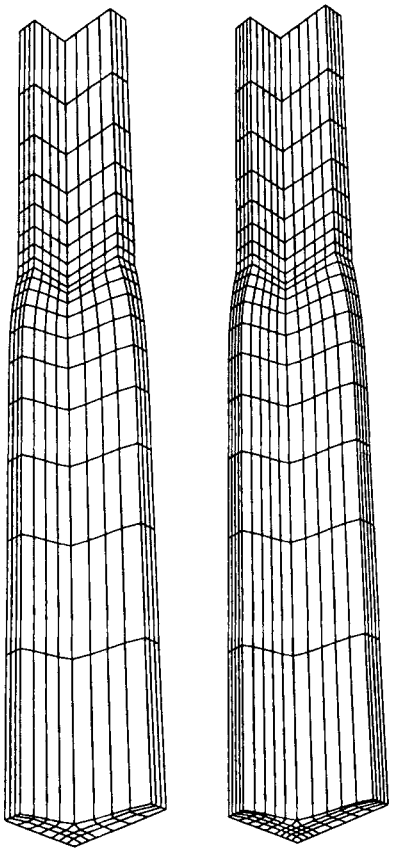

Figure 9. Perspective plot of the deformed meshes for the cross-like die

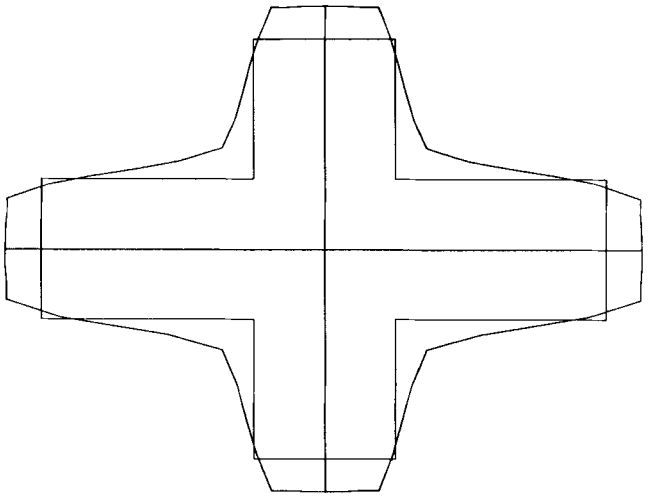

Figure 10. Comparison of the cross-like die section an extrudate profile

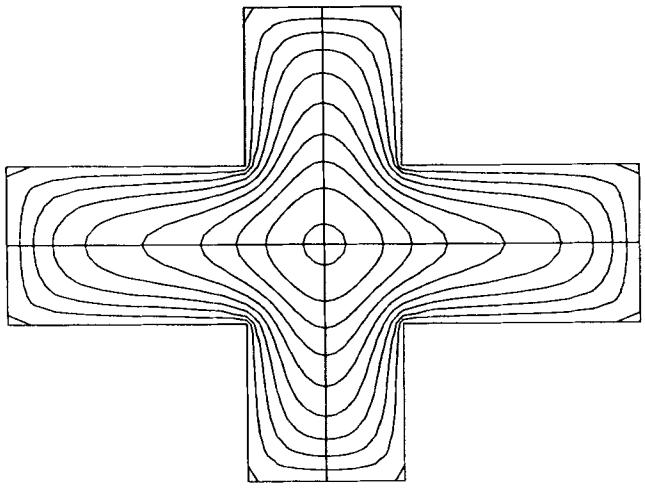

Figure 11. Velocity profile in the cross-like die

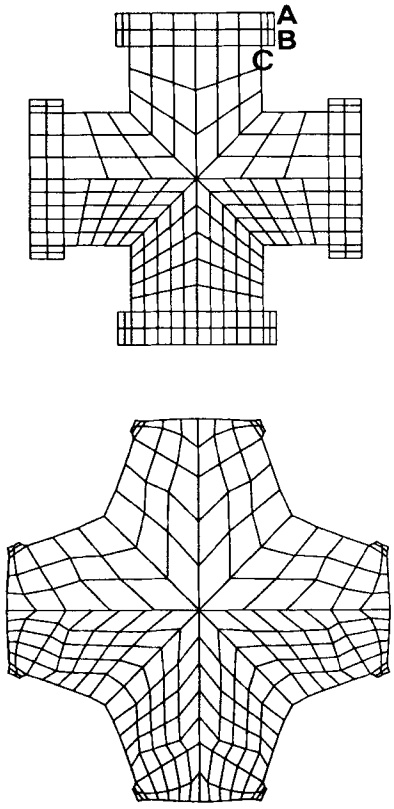


Figure 12. Profile with 28 corners: mesh refinement analysis

mesh. The agreement between these two solutions is excellent. A comparison between the fully developed extrudate profile and the die section is presented in Figure 10. Figure 11 presents contour lines of the fully developed normalized velocity profile.

Rearrangement of the velocity field causes large deformations in the extrudate, especially around the re-entrant corner B. Interestingly, these deformations are not caused by shear-thickening effects or by elasticity, as one could argue on the basis of two-dimensional extrusion simulations. As a reminder, the swelling ratio for a Newtonian fluid is 1·19 for a planar die and 1·13 for an axisymmetric die.

### *5.3. Profile with 28 corners*

Let us consider the die section described in Figure 12, which has four planes of symmetry. The use of normal–tangential boundary conditions allows us to reduce the computational domain to one-eighth of the section. The intersection between the mesh and the die exit plane is displayed in Figure 12. Discontinuity of the normal has been used for corners A, B and C and a remeshing rule based on the Euclidean distance has been used. As in previous examples, convergence of the iteration scheme has been obtained in five iterations. A comparison between the die section and the fully developed extrudate is shown in Figure 13. A perspective plot of the domain boundary is presented in Figure 14 (one-eighth of the domain). A comparison between two mesh refinements is shown in Figure 12.

The effects of the rearrangement of the velocity profile are even more pronounced than in the previous example. The angle of corner B has increased significantly. It must be noted that this region corresponds to important axial velocities in the die itself, as shown in Figure 15, which presents contour lines of the fully developed normalized velocity profile.

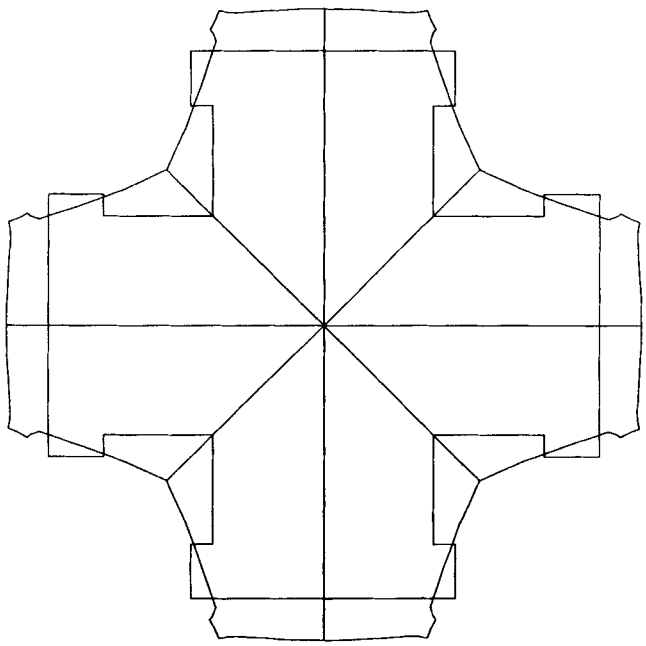

Figure 13. Comparison of the die with 28 corners and extrudate profiles

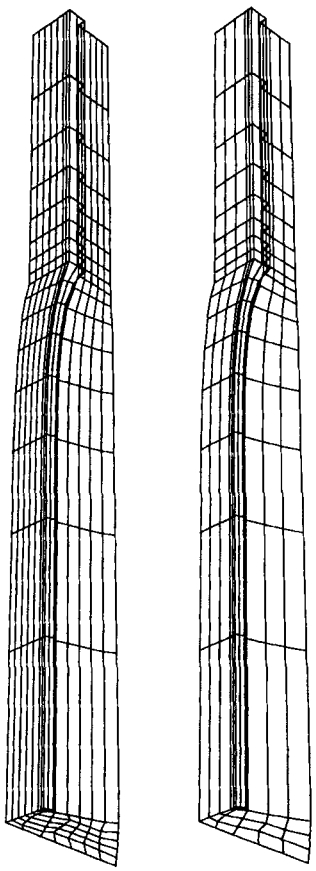

Figure 14. Perspective plot of the deformed meshes for the profile with 28 corners

## 6. CONCLUSIONS

We have introduced a velocity–pressure–position finite element technique to solve free surface flows in three-dimensional geometries. Discontinuity of the normal direction along corners has been used and a geometrically general remeshing technique based on the Euclidean distance has been introduced. A full Newton–Raphson scheme has been derived which allows us to compute the flow of a Newtonian fluid in a complex geometry in five iterations. For the problem of extrusion out of a rectangular die we have observed that no solution convergent with mesh refinement can be obtained if a single normal direction and a single kinematic condition are used for all nodes on the free surface.

Extrusion of profiles involving several corners has then been considered. We have computed the extrudate shapes out of a cross-like die and have shown that important deformations occur around re-entrant corners. These deformations are caused by a rearrangement of the non-uniform velocity profile in the die channel. Convergence of the solution with mesh refinement has been verified and the agreement on the extrudate shape was excellent.

Our last example is an extrudate profile involving 28 corners. As in previous simulations, the scheme converges quadratically to the solution and our remeshing algorithm is able to maintain the element deformation at an acceptable level. The extrudate shape exhibits large deformations

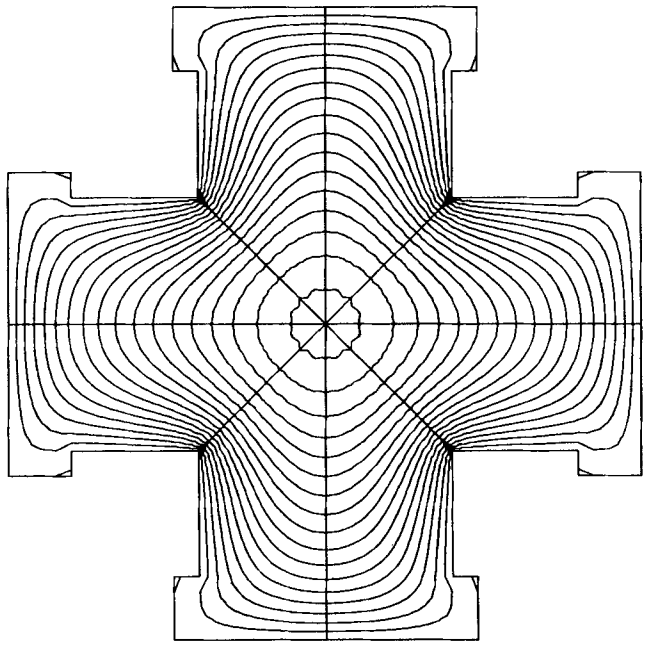

Figure 15. Velocity profile in the die with 28 corners

in the vicinity of all re-entrant corners, which could not be predicted on the basis of two-dimensional planar or axisymmetric simulations.

It is our intention to extend our technique to multifluid flows[10] in the near future. In this case distinct remeshing techniques in each fluid domain will be used. Early results show that the prediction of coupled interface and free surface positions is very promising.

ACKNOWLEDGEMENT

V. Legat wishes to acknowledge a scholarship from the Belgian 'Fonds National de la Recherche Scientifique'.

REFERENCES

1. M. B. Bush and N. Phan-Thien, 'Three-dimensional viscous flows with a free surface: flow out of long square die', *J. Non-Newtonian Fluid Mech.*, **18**, 211–218 (1985).
2. T. Tran-Cong and N. Phan-Thien, 'Three-dimensional extrusion process by boundary element method: II Extrusion of a viscoelastic fluid', *Rheol. Acta*, **27**, 639–648 (1988).
3. A. Karagiannis, A. N. Hrymak and J. Vlachopoulos, 'Three-dimensional extrudate swell of creeping Newtonian jets', *AIChE J.*, **34**, 2088–2094 (1988).
4. A. Karagiannis, A. N. Hrymak and J. Vlachopoulos, 'Three-dimensional non-isothermal extrusion flows', *Rheol. Acta*, **28**, 121–133 (1989).
5. H. Saito and L. E. Scriven, 'Study of coating flow by the finite element method', *J. Comput. Phys.*, **42**, 53 (1981).
6. O. A. Ladyzhenskaya, *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach, New York-London-Paris, 1969.
7. J. F. Thompson, 'Grid generation techniques in computational fluid dynamics', *AIAA J.*, **22**, 1505–1523 (1984).
8. P. Henrici, *Applied and Computational Complex Analysis, Vol. III.* John Wiley & Sons, New York-London-Sydney-Toronto, 1986.
9. O. Wambersie and M. J. Crochet, 'Pseudo-transient finite element method for calculating three-dimensional extrusion', *Int. j. numer. methods fluids*, **14**, 343–360 (1992).
10. A. Karagiannis, A. N. Hrymak and J. Vlachopoulos, 'Three-dimensional studies on bicomponent extrusion', *Rheol. Acta*, **29**, 71–87 (1990).